PKU-ICST at TRECVID 2015: Instance Search Task

Yuxin Peng, Jian Zhang, Xin Huang, Meng Sun,

Xiangteng He, Panpan Tang, Yunzhen Zhao,

Junjie Zhao, Jinwei Qi, and Junchao Zhang

Institute of Computer Science and Technology,

Peking University, Beijing 100871, China.

pengyuxin@pku.edu.cn

Abstract

We participated in all two types of instance search (INS) task in TRECVID 2015: automatic search and interactive search. This paper presents our approaches and results. In this task, we mainly focused on exploring the effective feature representation, feature matching and re-ranking algorithm. In this year, we also tried to use Deep Neural Networks (DNN) to improve the results. In feature representation, we extracted two kinds of features: (1) Bag-of-Words (BoW) feature based on Approximate K-means (AKM) and (2) DNN feature based on Convolutional Neural Networks (CNN). In feature matching, we adopted different ranking methods to different features: (1) For the AKM-based BoW feature, we used cosine distance to calculate the similarity between each query topic and each shot; (2) For the DNN feature, multi-bag SVM (MBSVM) was adopted since it can make full use of all query examples. Moreover, we conducted keypoint matching algorithm on the top ranked results. It was effective yet efficient since only top ranked results were considered. In re-ranking stage, we further incorporated transcripts into our framework to explore the context information. The official evaluations showed that our team is ranked 1st on both automatic search and interactive search.

1 Overview

In TRECVID 2015[6], we participated in all two types of Instance Search (INS) tasks: automatic search and interactive search. We totally submitted 4 runs including 3 runs for automatic search and 1 run for interactive search. The official evaluation results of our 4 runs are shown in Table 1.

In automatic search, our team is ranked 1st among all 13 teams. In interactive search, our team is also ranked 1st. Table 2 gives the detailed explanation of the brief descriptions in Table 1. Our system's framework is shown in Figure 1. Among the 3 automatic search runs, the difference between Run1 and Run3 is that Run1 uses early fusion strategy to combine different BoW features, while Run3 uses late fusion strategy. And Run3 is the fusion results of Run4 with DNN features.

Table 1: Results of our submitted 4 runs on Instance Search task of TRECVID 2015.

Type	ID	MAP	Brief description
Automatic	F_E_PKU_ICST_1	**0.4528**	K+D+M+R
	F_E_PKU_ICST_3	0.4433	K+D+M+R
	F_A_PKU_ICST_4	0.4242	K+M+R
Interactive	I_E_PKU_ICST_2	**0.5170**	K+D+M+R+H

Table 2: Description of our methods.

Abbreviation	Description
K	**K**eypoint based feature based on AKM, 1,000,000D
D	**D**eep Neural Networks based feature
M	Keypoint **M**atching
R	**R**e-ranking based on text matching
H	**H**uman feedback

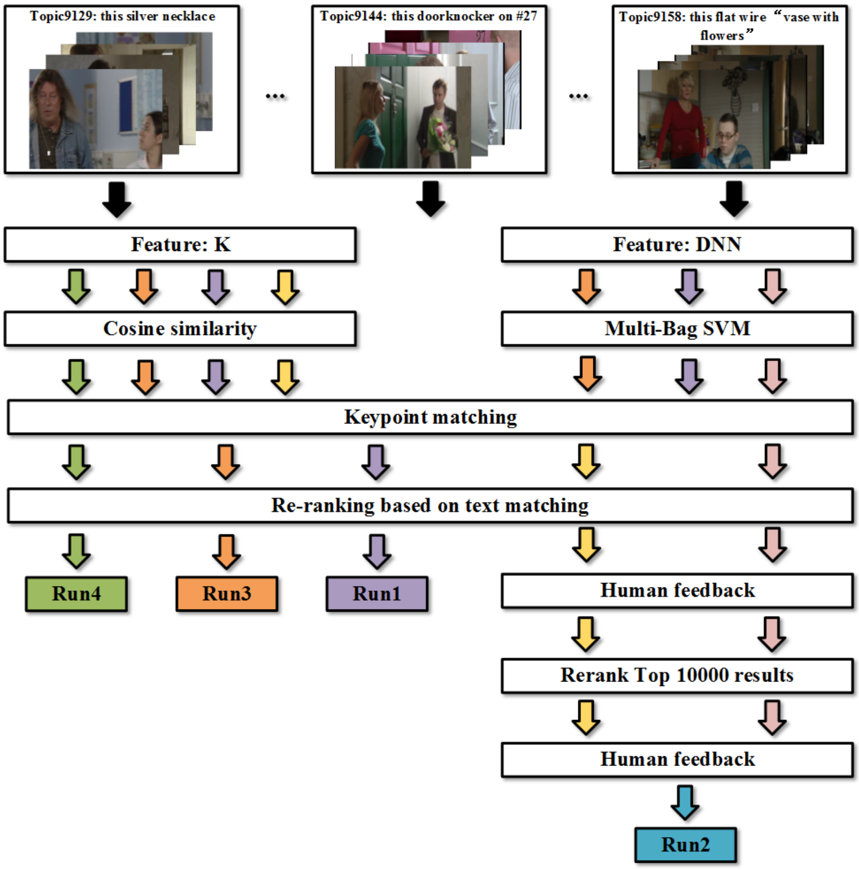
Topic9129: this silver necklace
Topic9144: this doorknocker on #27
Topic9158: this flat wire “vase with flowers”
…
…
Feature: K
Feature: DNN
Cosine similarity
Multi-Bag SVM
Keypoint matching
Re-ranking based on text matching
Run4
Run3
Run1
Human feedback
Rerank Top 10000 results
Human feedback
Run2

Figure 1: Framework of our instance search approach for the submitted 4 runs.

2 Feature Representation

We used two kinds of features for the instance search task, namely AKM-based BoW feature and DNN feature respectively.

2.1 AKM-based BoW features

We explored different keypoint-based BoW features to represent each video shot. In our method, the extraction of keypoint-based BoW features includes three steps:

(1) Firstly, we detected the keypoints using four different detectors from the keyframes, and used two descriptors to represent the neighboring regions around those keypoints.

(2) Secondly, we used AKM algorithm to cluster the keypoints into one-million clusters, and constructed a visual vocabulary with the cluster centroids.

(3) Thirdly, we quantized each shot into a BoW feature by assigning the keypoints of all frames in the shot to multiple nearest visual words (centroids), where the word weights were determined by the keypoint-to-word similarity and region of interest (ROI). Thus each shot could be represented by a one-million dimensional BoW feature. Therefore we totally obtained 4×2=8 BoW features for each video shot and each query topic.

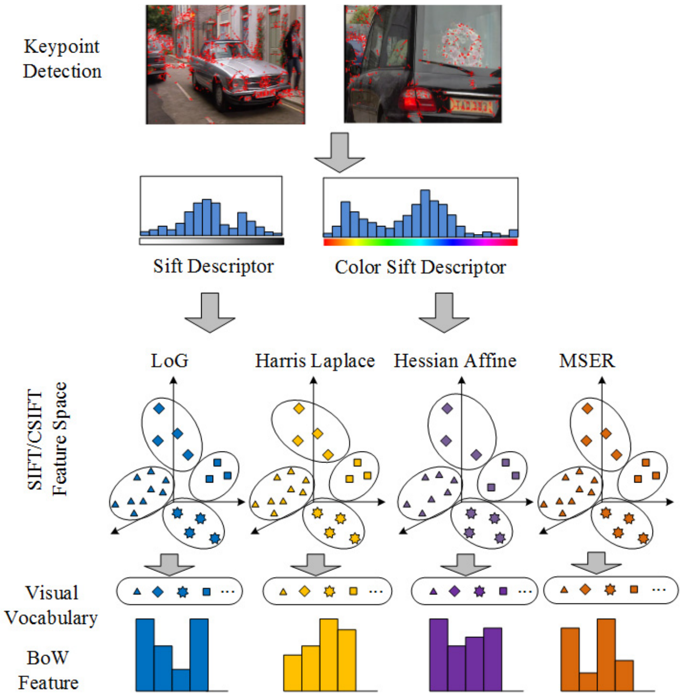

Figure 2: Combination of BoW features based on different detectors and descriptors.

In step (1), we adopted four complementary detectors to detect the keypoints from keyframes: Laplace of Gaussian (LoG)[1], Harris Laplace[2], Hessian Affine[3], and MSER [4]. For each detector, we used two descriptors to generate two BoW features: 128 dimensional SIFT descriptor[1] and 192 dimensional ColorSIFT descriptor[5]. As shown in Figure 2, for each combination of detector and descriptor, a one-million dimensional BoW feature was generated separately.

2.2 DNN feature

In this year, we also tried to adopt the DNN feature to the Instance Search task. We constructed a feature extraction framework based on two off-the-shell deep learning models, AlexNet[7] and VGGNet[8], both of which were convolutional neural networks. Our framework included two phases.

(1) CNN Model training

To adapt the two off-the-shell models to the INS task, we fine-tuned the CNN models by using INS task dataset. We made the training dataset from the query topic images and treated the topic categories as image annotations. Thus we got two INS-specific deep CNN models, one of which was AlexNet, and the other was VGGNet.

(2) Feature Extraction

Once we trained two CNN models, we considered them as two feature extractors. When using CNN as feature extractor, the activations of first fully-connected layer were outputted as features. For each keyframe image, we extracted two 4096 dimensional feature vectors based on these two CNN models respectively. Then two feature vectors were concatenated as the final feature vector, so we got an 8192 dimensional feature vector for each video shot and each query topic.

3 Feature Matching

In feature matching, we adopted two kinds of methods for different features. For the AKM-based BoW features, we used cosine distance to calculate the similarity between the query and each shot. We used both early and late fusion strategies on 8 BoW features, and the results showed that early fusion strategy achieved better results. For the DNN feature, multi-bag SVM (MBSVM) was adopted since it can make full use of all query examples. Moreover, we conducted keypoint matching algorithm on the top ranked results. It was very effective yet efficient since only top ranked results are considered.

The query examples were considered as positive samples. Due to the fact that only a few shots were relevant with the topics in the test data set, we adopted the random sampling of test data as negative examples. A problem of learning-based method was that there were too few positive samples compared to the many negative samples. In our approach, we used MBSVM algorithm to handle this imbalanced learning problem. The algorithm details are presented in Figure 3 and the diagram is shown in Figure 4.

(1) Over-sample the positive samples: Duplicate the positive sample set P for $(PCopy - 1)$ times and get a new set of positive samples P' with $PCopy \times PN$ samples, where *PN* is the number of positive samples in P before over-sampling.

(2) Under-sample the negative samples: Randomly select $NPR \times PCopy \times PN$ negative samples, and combine them with the over-sampled positive sample set P' to form a bag. That is to say, in each bag, the number of negative samples is *NPR* times as the number of positive samples, where *NPR (negative-to-positive-ratio)* is a parameter to control the degree of data imbalance in each bag. A model is trained by *LibSVM* for each bag, where *RKF* kernel is used with default parameters.

(3) Repeat the above step (2) for *BagNum* times, where *BagNum* is a parameter specifying the number of bags. Then for each shot in the test data set, the *BagNum* prediction scores given by different models are averaged to form the final result. Notice that the negative samples in each bag are selected without repetition, that is, the negative samples are totally different in these bags. This ensures that we can make full use of the most of negative samples.

Figure 3: our algorithm for learning-based retrieval.

Totally, there were three important parameters in MBSVM algorithm: *PCopy*, *NPR* and *BagNum*.–Experiments show that *PCopy=100*, *NPR=5* and *BagNum=5* could achieve good performance in both accuracy and efficiency, while *PCopy* needed to be set according to the number of frames extracted from each shot in the query examples.

We used keypoint matching method based on SIFT descriptor and keypoint mismatching elimination method based on RANSAC to further improve the performance. Since keypoint matching was time consuming, we only conducted keypoint matching and mismatching-elimination algorithm on the top 1000 ranked shots, which was effective yet efficient.

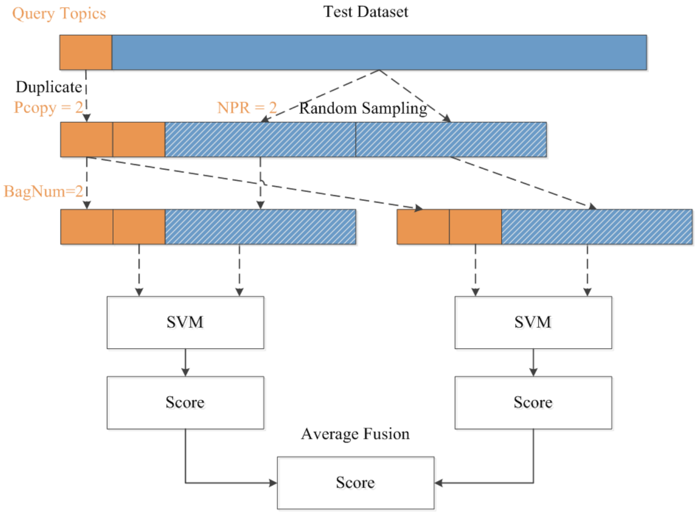

Figure 4: Diagram of MBSVM algorithm, where Pcopy=2, NPR=2 and BagNum=2.

4 Re-ranking

In re-ranking stage, NIST provides the transcripts about the videos this year, and some topics explicitly point out the name of the instance to search, such as the topic 9139 ("this shaggy dog (Genghis)") and the topic 9142 (this chihuahua (Prince)). An instance appeared in the corresponding shot when its name appeared in the transcript. We re-ranked such shot to the top of the ranking list. Also according to the transcripts which contained the name of the instances, we found some words that appeared frequently with the name automatically, and these words were also used for query expansion.

5 Interactive Search

This year we adopted a two-turn interactive search process. When given the retrieved results by automatic search (F_E_PKU_ICST_1), we preserved the top 10000 results for re-ranking, and showed users 1000 video shots with the highest scores. Then users manually selected several positive samples as expanded queries. For the diversity of information, users were encouraged to find shots with relatively clear difference. Then we measured the visual similarities between the top 10000 results and the expanded queries. The final similarity scores were obtained as a late fusion of the scores computed by the original and expanded queries. According to the fused visual similarities, we got a new ranking list of the top 10000 results. Now the top 1000 results were shown to users again, and the manual re-ranking was performed. As shown in table 1, the interactive process significantly improved the search accuracy.

6 Conclusion

By participating in the instance search task in TRECVID 2015, we have the following conclusions: (1) Effective features are still vital, (2) DNN feature is complementary with keypoint-based features, since DNN feature can be used to retrieve different positive results, and (3) The keypoint matching is very helpful.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grants 61371128 and 61532005, National Hi-Tech Research and Development Program of China (863 Program) under Grant 2014AA015102, and Ph.D. Programs Foundation of Ministry of Education of China under Grant 20120001110097. We also thank BBC for providing the EastEnders dataset: Programme material copyrighted by BBC.

References

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision (IJCV)*, vol. 60, no.2, pp. 91-110, 2004.

[2] C.G.M. Snoek, K.E.A. van de Sande, O. de Rooij, et al., “The MediaMill TRECVID 2008 Semantic Video Search Engine”, *TRECVID 2008 Workshop*, NIST, USA, 2008.

[3] K. Mikolajczyk, and C. Schmid, “Scale and affine invariant interest point detectors”, *International Journal of Computer Vision (IJCV)*, vol. 60, no. 1, pp. 63-86, 2004.

[4] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions”, *British Machine Vision Conference (BMVC)*, pp. 384-393, 2002.

[5] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 27, no.10, pp. 1615-1630, 2004.

[6] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, G. Quéenot and R. Ordelman, “TRECVID 2015 -- An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics”, *TRECVID 2015 Workshop*, NIST, USA, 2015.

[7] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks”, *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097-1105, 2012.

[8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, arXiv preprint arXiv:1409.1556, 2014.